Atlas LSH Neural Networks:

Persistent Global Context as a Mechanism for Conditional Computation

Executive Summary

Conventional neural networks progressively transform internal representations as information flows through depth. Although activations vary from input to input, the network architecture and parameter set are largely fixed.

Atlas LSH neural networks introduce an additional source of information: a persistent global context signal derived from a locality sensitive hash (LSH) of the input.

This context signal is presented repeatedly to the network throughout depth and determines which parameters or operators are selected at each layer.

Viewed in this way, Atlas architectures are not simply deep neural networks. They are systems that perform conditional computation under continuous guidance from an externally derived geometric description of the input.

---

1. Conditional Computation in Conventional Networks

A standard feedforward network computes

$$h_{l+1}=\sigma\left(W_l h_l + b_l\right)$$

where:

- h_l is the hidden representation at layer l,
- W_l is a fixed parameter matrix,
- σ is a nonlinearity such as ReLU.

Although activations change from input to input, the network itself remains fixed after training.

ReLU networks do introduce a limited form of conditional computation. Neurons whose pre-activations are negative are effectively removed from the computation.

For a given input, different subnetworks become active.

However, these decisions are:

- local,
- generated internally,
- based only on current hidden activations.

---

2. Atlas LSH Networks Introduce Persistent Context

Atlas LSH neural networks begin by computing a locality sensitive hash code

$$b(x) \in \{0,1\}^k$$

from the input x.

The hash bits are typically derived from random projections:

$$b_i(x) = \text{sign}(r_i^T x)$$

where r_i is a random projection vector.

Each bit therefore indicates on which side of a random hyperplane the input lies.

The collection of bits provides a coarse geometric address for the input.

The key idea is simple:

> The hash code is made available throughout the network and is repeatedly used to select computations.

---

3. Two Information Streams

Atlas networks may be viewed as containing two distinct information streams.

Representation stream

The conventional hidden representation evolves through depth:

input → h1 → h2 → h3 → ... → output

This stream describes:

> What has been computed so far.

Context stream

The LSH code is supplied repeatedly:

LSH(x) ——————————————————————→ every layer

This stream describes:

> Where the input lies in the Atlas.

Each layer therefore computes

$h_{l+1} = F_l(h_l, b(x))$.

The network no longer relies solely on internal activations when deciding what computation to perform.

---

4. Persistent Global Context

The repeated use of LSH bits creates a persistent global context signal.

Unlike conventional networks, information about the original input geometry is continuously available at every depth.

Each layer effectively receives two pieces of information:

1. the current hidden state,
2. the global context describing the input region.

The network repeatedly asks:

> Given what has already been computed, and given where this input lies in input space, what computation should be performed next?

This persistent context signal may provide several benefits.

Possible consequences include:

- context-dependent computation,
- improved specialization,
- stronger conditional computation,
- preservation of global geometric information,
- reduced loss of input-specific context at great depth.

---

5. Atlas Coordinates

The LSH bits may be interpreted as coordinates within a computational Atlas.

Nearby inputs tend to share many bits.

Consequently:

- nearby inputs tend to share computational routes,
- distant inputs tend to follow different routes.

The Atlas therefore partitions input space into regions and associates different computations with different regions.

The network becomes a collection of reusable computational operators indexed by location within this Atlas.

---

6. Repeated Conditionality Through Depth

Suppose each layer receives n bits of context.

Across L layers, conditional information is introduced repeatedly.

The effective computation becomes

$W_L(b_L)W_{L-1}(b_{L-1}) \dots W_1(b_1)$.

Even simple parameter selection schemes can produce an enormous number of possible computational pathways.

The repeated injection of context therefore creates a highly expressive conditional computation system.

Importantly, this conditionality is externally derived rather than internally generated.

---

7. Comparison with Conventional Networks

Property	**Conventional ReLU Network**	**Atlas LSH Network**
Conditionality source	Hidden activations	Persistent hash context
Decision scope	Local	Global
Context availability	Implicit	Explicit
Parameter selection	Limited	Extensive
Input geometry access	Gradually transformed	Continuously available

Conventional networks progressively transform representations.

Atlas networks progressively transform representations while continuously consulting a global context signal.

---

8. A Conceptual Interpretation

One possible interpretation is:

> Conventional neural networks learn a fixed computational circuit.

Atlas LSH neural networks learn a large collection of reusable computational operators together with rules for selecting and composing those operators based on the geometric location of the input.

The network therefore behaves less like a fixed circuit and more like a computation guided by a continuously available map.

Conclusion

Atlas LSH neural networks can be viewed as architectures that implement conditional computation through the repeated injection of a persistent global context signal.

The hidden activations describe what has already been computed.

The LSH bits describe where the input lies in the computational Atlas.

Together these two information streams determine the computation performed at every layer.

This separation between representation and context may provide a useful framework for understanding and designing future conditional-computation architectures.